CLAIMS

1. A method, comprising:

executing a program in a logical address space of a computer, with an address translation circuit translating address references generated by the program from the program's logical address space to the computer's physical address space;

recording profile information that records physical memory addresses referenced during an execution interval of the program.

2. The method of claim 1, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references records the event of a sequential execution flow across a page boundary in the address space.

3. The method of claim 1, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references is a divergence from sequential execution flow consequent to recognizing or handling of an external interrupt.

4. The method of claim 1, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer, and at least one of the recorded instruction references indicates the address of the last byte of an instruction executed by the computer during the profiled execution interval.

5. The method of claim 4:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising recording profile information describing the processor mode during the profiled execution interval, the profile information being efficiently tailored to

annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

6. The method of claim 4:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

7. The method of claim 4, the recorded profile information being efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

8. The method of claim 1:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising recording profile information describing a processor mode during the profiled execution interval, the profile information being efficiently tailored to annotate the profiled binary code with sufficient processor mode information to resolve mode-dependency in the binary coding.

9. The method of claim 8:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

10. The method of claim 9, wherein the triggering event is expiration of a timer.

11. The method of claim 8, wherein:

the recorded profile information describes at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction;

the recorded profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

12. The method of claim 8, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing

the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

13. The method of claim 8, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

14. The method of claim 8, the recorded profile information being efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

15. The method of claim 8, wherein an individual record of a recorded profileable event includes an event code describing a class of the profileable event recorded in the record, a number of bits used to record the event code being less than $\log_2$ of the number of event classes distinguished by the event code.

16. A method, comprising:

executing a program on a computer, the program referring to memory by virtual address; and

concurrently with the execution of the program, recording profile information describing memory references made by the program, the profile information recording physical addresses of the profiled memory references.

17. The method of claim 16:

wherein the program has been compiled without special consideration for execution profiling;

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

commencing the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing every event that matches the profileable event selection criteria induced after the triggering event, the recording continuing until a predetermined stop condition is reached.

18. The method of claim 17:

wherein an interpretation of instructions of the program depends on a processor mode not expressed in a binary representation of the instruction;

the recorded profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction;

the recorded profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

19. The method of claim 17, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

20. The method of claim 19, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

21. The method of claim 17, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

22. The method of claim 17, wherein the triggering event is the expiration of a timer.

23. The method of claim 16, wherein:

the program has been compiled without special consideration for execution profiling; and

at least one of the recorded physical memory references notes the source and destination of a control flow event in which control flow of the program execution diverges from sequential execution.

24. The method of claim 23:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

the recorded profile information satisfying at least two of the following three properties:

the recorded profile information describes all divergences of execution from sequential execution occurring during the profiled execution interval;

the recorded profile information describes all processor mode changes not inferable from the opcode of the instruction, taken together with a processor mode before the mode change instruction occurring during the profiled execution interval, that induces the processor mode change;

the recorded profile information identifies each distinct physical page of instruction text executed during the profiled execution interval.

25. The method of claim 23, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

26. The method of claim 23, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

27. The method of claim 23, the recorded profile information being efficiently tailored to identify all bytes of object code executed during the profiled execution interval, without reference to the binary code of the program.

28. The method of claim 23, further comprising:

recording profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program.

29. The method of claim 23, wherein an individual record of a recorded profileable event includes an event code describing a class of the profileable event recorded in the record, a number of bits used to record the event code being less than $\log_2$ of the number of event classes distinguished by the event code.

30. The method of claim 16:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising, recording profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

31. The method of claim 30, wherein the divergence from sequential execution flow is consequent to recognizing or handling of an exception.

32. The method of claim 30, wherein:

an instruction of the computer, having a primary effect on the execution the computer not related to profiling, has an immediate field for an event code encoding the nature of a profileable event to be recorded in the profile information, the immediate field having no effect on computer execution except to determine the event code of the profiled event.

33. The method of claim 30, further comprising:

recording profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program.

34. The method of claim 16:

wherein the program has been compiled without special consideration for execution profiling;

the profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

35. The method of claim 34, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

36. The method of claim 34, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

37. The method of claim 34, wherein distinct physical pages are detected consequent to sequential execution flow in the logical address space of the program across a page boundary of the address space.

38. The method of claim 16:

wherein the program has been compiled, without special consideration for execution profiling, into an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction;

and further comprising, recording profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, taken together with a processor mode before the mode change instruction;

the profile information further identifying each distinct physical page of instruction text executed during the profiled execution interval.

39. The method of claim 38, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

40. The method of claim 39, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

41. The method of claim 38, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

42. The method of claim 38, wherein:

an instruction of the computer, having a primary effect on the execution the computer not related to profiling, has an immediate field for an event code encoding the nature of a profileable event to be recorded in the profile information, the immediate field having no effect on computer execution except to determine the event code of the profiled event.

43. The method of claim 16, wherein the program is executed on a computer having:

an instruction pipeline configured to execute instructions from a memory of the computer; and

profile circuitry configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

44. The method of claim 43, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

45. The method of claim 43, further comprising:

recording profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program.

46. The method of claim 43:

wherein the program execution induces occurrence of events that match time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program;

and further comprising, during a profile-quiescent interval of execution, recording no profile information in response to the occurrence of profileable events; and

initiating the profiled execution interval after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the recorded profile information describing the profileable events induced after the triggering event.

47. The method of claim 43, wherein:

an instruction of the computer, having a primary effect on the execution the computer not related to profiling, has an immediate field for an event code encoding the nature of a profileable event to be recorded in the profile information, the immediate field having no effect on computer execution except to determine the event code of the profiled event.

48. The method of claim 43, further comprising:

when an instruction fetch during the profiled execution interval causes a miss in a translation look aside buffer (TLB), the instruction fetch triggering a profileable event, servicing

the TLB miss and reflecting the corrected state of the TLB in the profile information recorded for the profileable instruction.

49. The method of claim 43, further comprising injecting multiple consecutive operations into the instruction pipeline to capture a single profileable event.

50. The method of claim 43, wherein the profile circuitry injects an instruction into the pipeline, the instruction controlling the pipeline to cause the profileable event to be materialized in an architecturally-visible storage register of the computer.

51. The method of claim 16, further comprising the steps of:

on a first CPU of a multiprocessor computer, executing a program, and collecting profile data describing the execution of the program;

on a second CPU of the multiprocessor, while the profiling of the program continues, analyzing the collected profile data;

controlling the execution of the program on the first CPU based at least in part on the analysis of the collected profile data.

52. The method of claim 51, further comprising:

recording profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program.

53. The method of claim 16, wherein:

the profile information is recorded into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

54. The method of claim 53, further comprising:

recording the profile information into a general register indicated by a register pointer; and

incrementing the register pointer to a next general register in which to record next profile information, without software intervention.

55. The method of claim 54, further comprising:

detecting when a range of general registers available for collecting profile information is exhausted; and

storing the profile information from the general registers to a main memory of the computer when exhaustion is detected.

56. The method of claim 55, further comprising:

generating an exception on the detection of exhaustion, and performing the storing in a handler for the exception.

57. The method of claim 53, further comprising:

when an instruction fetch during the profiled execution interval causes a miss in a translation look aside buffer (TLB), the instruction fetch triggering a profileable event, servicing the TLB miss and reflecting the corrected state of the TLB in the profile information recorded for the profileable instruction.

58. The method of claim 16, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer.

59. The method of claim 58, wherein at least one of the recorded instruction references is a sequential execution flow across a page boundary in the address space.

60. The method of claim 59, wherein at least one of the recorded instruction references is a sequential execution flow across a page boundary in the address space occurring within a single instruction.

61. The method of claim 59, wherein at least one of the recorded instruction references is a sequential execution flow across a page boundary in the address space between two instructions that are sequentially adjacent in the logical address space.

62. The method of claim 58, at least one of the recorded instruction references is a divergence of control flow consequent to an external interrupt.

63. The method of claim 16, further comprising:

recording profile information recording a processor mode that determines the meaning of binary instructions of the computer.

64. The method of claim 16, further comprising:

recording profile information recording a data-dependent change to a full/empty mask for registers of the computer.

65. The method of claim 16, further comprising:

recording profile information that records a sequence of events of the program during the profiled execution interval, the sequence including every event that matches time-independent criteria of profileable events to be profiled, the criteria including at least some physical memory references referenced by the program.

66. The method of claim 16, wherein:

an instruction of the computer, having a primary effect on the execution the computer not related to profiling, has an immediate field for an event code encoding the nature of a profileable event to be recorded in the profile information, the immediate field having no effect on computer execution except to determine the event code of the profiled event.

67 A computer, comprising:

an instruction pipeline and memory access unit configured to execute instructions in a logical address space of a memory of the computer;

an address translation circuit for translating address references generated by the program from the program's logical address space to the computer's physical address space; and

profile circuitry cooperatively interconnected with the instruction pipeline and configured to detect, without compiler assistance for execution profiling, occurrence of profileable events occurring in the instruction pipeline, and cooperatively interconnected with the memory access unit to record profile information describing physical memory addresses referenced during an execution interval of the program.

68. The computer of claim 67, wherein the recorded physical memory references include addresses of binary instructions referenced by an instruction pointer during a sequential execution flow across a page boundary in the address space.

69. The computer of claim 67, wherein at least one of the recorded instruction references indicates the address of the last byte of an instruction executed by the computer during the profiled execution interval.

70. The computer of claim 67 wherein:

during a profile-quiescent interval of execution of a program that has been compiled without special consideration for execution profiling and that induces events that match time-independent criteria of profileable events to be profiled, the selection criteria including at least some references to the physical address space, the profile circuitry is configured to record no profile information in response to the occurrence of profileable events; and

after a triggering event is detected, the triggering event being one of a predefined class of triggering events, the profile circuitry is configured to commence the profiled execution interval and to record profile information describing every event that matches the profileable event selection criteria induced during the profiled execution interval, the recording continuing until a predetermined stop condition is reached.

71. The computer of claim 67, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution, and

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction.

72. The computer of claim 71, wherein:

the profile circuitry is configured to detect the occurrence of profileable events occurring in the instruction pipeline, and to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

73. The computer of claim 71, wherein:

the profile circuitry is configured to record profile information into general registers of the computer under hardware control, without software intervention, and without first storing the profile information into main memory.

74. The computer of claim 67, wherein:

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

75. The computer of claim 67, wherein:

the instruction pipeline is configured to execute instructions of an instruction set in which an interpretation of an instruction depends on a processor mode not expressed in a binary representation of the instruction; and

the profile circuitry is configured to record profile information describing at least all events occurring during the profiled execution interval of the two classes:

a divergence of execution from sequential execution;

a processor mode change that is not inferable from the opcode of the instruction that induces the processor mode change, the instruction opcode taken together with a processor mode before the mode change instruction; and

the profile circuitry is configured to record profile information identifying each distinct physical page of instruction text executed during the profiled execution interval.

76. The computer of claim 67, wherein:

the profile circuitry is configured to direct the instruction pipeline to record profile information describing the profileable events essentially concurrently with the occurrence of the profileable events, the detecting and recording occurring under hardware control without software intervention.

77. The computer of claim 76, wherein:

the profile circuitry is configured to direct the instruction pipeline to record the profile information into general registers of the computer, without software intervention, and without first storing the profile information into main memory.

78. The computer of claim 67, wherein:

the profile circuitry is configured to direct the instruction pipeline to record the profile information into general registers of the computer, without software intervention, and without first storing the profile information into main memory.

79. The computer of claim 67, further comprising:

profile control bits implemented in the computer hardware, values of the profile control bits controlling a resolution of the operation of the profile circuitry;

a binary translator configured to translate programs coded in a first instruction set architecture into instructions of a second instruction set architecture;

a profile analyzer configured to analyze the recorded profile information, and to set the profile control bits to values to improve the operation of the binary translator.

80. The computer of claim 67:

wherein the instruction pipeline is configured to execute instructions of two instruction sets, a native instruction set providing access to substantially all of the resources of the computer, and a non-native instruction set providing access to a subset of the resources of the computer.

81. The computer of claim 80, wherein the instruction pipeline and profile circuitry are further configured to effect recording of profile information describing an interval of the execution of an operating system coded in the non-native instruction set.